

Защита в операционных системах

Вязанкин Олег Николаевич

к.ф.-м.н., доцент кафедры ПОЗИ, ИМИТ

ovyazankin@gmail.com



Тема 4. Аутентификация.



План

- Общие сведения
- Аутентификация в UNIX
- Аутентификация в Windows

Общие сведения

- *Идентификация* субъекта доступа заключается в том, что субъект сообщает системе *идентификационную информацию* о себе, и таким образом идентифицирует себя.
- *Аутентификация* субъекта доступа заключается в том, что субъект предоставляет системе помимо идентификационной информации еще и *аутентификационную информацию*, подтверждающую, что он действительно является тем субъектом доступа, к которому относится идентификационная информация.
- *Авторизация* субъекта доступа происходит после успешной идентификации и аутентификации. При авторизации субъекта ОС выполняет действия, необходимые для того, чтобы субъект мог начать работу в системе – загружает индивидуальные настройки пользователя, запускает программу-оболочку и т.п.

Общие сведения

Аутентификация может осуществляться как для физических пользователей, так и для псевдопользователей. Однако, наибольший интерес с точки зрения обеспечения информационной безопасности представляет аутентификация физических пользователей.

В системе с адекватной политикой безопасности физический пользователь просто не может войти в систему от имени псевдопользователя. Поэтому в дальнейшем мы будем рассматривать аутентификацию только обычных пользователей.

Обычно подсистема аутентификации ОС строится по одной из трех следующих схем:

- парольная аутентификация;
- аутентификация с использованием внешних носителей информации;
- биометрическая аутентификация.



Общие сведения

Парольная аутентификация

- Имя пользователя – как правило, назначается ему администратором системы.
- Пароль – текстовая строка, известная только пользователю.
- Администратор не должен знать пароль пользователя. Это может привести к компрометации пользователя, что недопустимо в защищенной системе.
- Из этого следует, что пароль не должен храниться в системе в открытом виде.

Общие сведения

Парольная аутентификация

Обычно для шифрования паролей в списке пользователей применяют одну из известных *криптографически стойких хеш-функций* – функций, обладающих следующими тремя свойствами:

- сложность вычисления функции $h(X)$ линейно зависит от размерности входа;
- сложность вычисления обратной функции $h^{-1}(Y)$ экспоненциально зависит от размерности входа;
- все значения функции равномерно распределены, нет аномальных значений, встречающихся много реже или много чаще, чем другие.

При применении для шифрования паролей криптографически стойкой хеш-функцией в списке пользователей хранится не сам пароль X , а *образ пароля* $h(X)$.

Однонаправленность хеш-функции не позволяет восстановить пароль по образу пароля, но позволяет, вычислив хеш-функцию, получить образ введенного пользователем пароля и, сравнив его с эталонным образом пароля, проверить правильность введения пароля.

Общие сведения

Парольная аутентификация

Несмотря на криптографическую стойкость образов паролей, образы паролей не должны быть общедоступны. Хранение образов паролей в файле или базе данных, к которой имеют доступ только системные процессы, создает дополнительный редут защиты, которым не следует пренебрегать.

В процедуре генерации образа пароля обязательно должен участвовать *маркант* (*криптографическая соль*) – данные, генерируемые случайным образом, не являющиеся секретом и хранящиеся в открытом виде вместе с образом пароля. Другими словами, вместо $h(X)$ в системе должна храниться пара $(M, h(X, M))$. Это необходимо для того, чтобы одинаковым паролям разных пользователей соответствовали разные образы и, кроме того, это затрудняет атаку на пароль по словарю.

Общие сведения

Парольная аутентификация

Пример.

Длина случайной строки (соли) - 8 символов (буквы латинского алфавита (52) и цифры (10)).

Во сколько раз увеличится размер словаря для хранения хеш-значений и исходных слов? В $62^8 = 218340105584896$ раз.

Допустим, исходный словарь занимал 1 ГБ дискового пространства. При использовании соли понадобится размер дискового пространства $218340105584896 * 1024 * 1024 * 1024 = 2.344 * 10^{23}$ байт.

Если мы располагаем технологией создания дисков размером в 1 ПБ (10^{15} байт), нам понадобится таких дисков в количестве более чем 2 миллиона!



Общие сведения

Парольная аутентификация

Если пользователь, входящий в систему, неправильно ввел свое имя и пароль, система должна выдать ему сообщение об ошибке, не указывая, какая именно информация некорректна. В противном случае подбор пароля существенно упрощается.

Для системы парольной аутентификации существует две основные угрозы:

- компрометация пароля,
- подбор пароля.

Общие сведения

Парольная аутентификация

Для обеспечения надежной защиты от компрометации паролей подсистема должна удовлетворять следующим требованиям:

- пароль, вводимый пользователем, не отображается на экране компьютера;
- ввод пароля из командной строки недопустим.

Кроме того, пользователи должны быть проинструктированы о:

- необходимости хранения пароля в тайне от других пользователей, включая администратора системы;
- необходимости немедленной смены пароля после его компрометации;
- необходимости регулярной смены пароля;
- недопустимости записи пароля на бумагу или в файл.

Общие сведения

Парольная аутентификация

Что касается подбора паролей, прежде чем перейти к описанию средств защиты от этой угрозы, сначала рассмотрим более подробно методы подбора паролей.

Чаще всего применяется так называемый оффлайн-подбор, осуществляемый вне атакуемой системы. В этом случае нарушитель должен предварительно получить из атакуемой системы хеш-образы подбираемых паролей.

При оффлайн-подборе паролей применяются следующие методы:

- тотальный перебор (brute force – метод грубой силы);
- подбор пароля по словарю;
- подбор пароля с использованием знаний о пользователе;
- подбор образа пароля.

Общие сведения

Парольная аутентификация

Список некоторых брутеров:

- John the Ripper (<http://www.openwall.com/john/>),
- MDCrack (<http://c3rb3r.openwall.net/mdcrack/>),
- MD5Inside (<http://www.insidepro.com/>),
- PasswordsPro (<http://www.insidepro.com/>),
- UDC (<http://the-udc.com/>).

Общие сведения

Парольная аутентификация

Существует целый ряд методов, позволяющих несколько уменьшить угрозу компрометации и подбора паролей пользователей. Вот некоторые из них:

- ограничение срока действия пароля,
- ограничения на пароль (длина, набор символов и т.д.),
- блокировка терминала,
- блокировка учетной записи пользователя,
- ограничения на режим входа в систему пользователей, пользующихся плохими паролями,
- генерация паролей системой,
- некоторые из перечисленных методов могут применяться в совокупности.

Общие сведения

Парольная аутентификация

Оценим какой должна быть оптимальная длина пароля.

Для начала предположим, что используется пароль в 10 символов и множество допустимых символов включает буквы латинского алфавита, цифры и 10 специальных символов.

Далее будем считать, что на одном компьютере мы можем подсчитать за 1 секунду 10^9 хеш-значений. Тогда при тотальном переборе мы можем определить пароль за время $72^{10} / 10^9$ в секундах. В пересчете на дни мы получим следующую оценку:

$72^{10} / 10^9 / 3600 / 24 = \mathbf{43332.25}$ дней.

Если злоумышленник будет использовать ботнет из, например, 100000 компьютеров, то на взлом пароля уйдет меньше одного дня.

Увеличим длину пароля на 2 символа. Тогда время взлома пароля возрастет до 224634374.55 дней для одного компьютера, а для сети из 100000 компьютеров понадобится примерно **2246** дня. Даже если предположить, что с вероятностью 50% пароль будет найден за половину времени (1123 дня), вряд ли это устроит злоумышленника.

Можно также заметить, что в последние годы получили распространение искусственно усложненные хеш-функции, для которых вычисление каждого значения занимает на современных процессорах несколько миллисекунд или даже секунд.



Общие сведения

Аутентификация с использованием внешних носителей информации

При таком способе аутентификации аутентификационная информация хранится на внешнем носителе информации (пластиковая карта, touch memory, электронный ключ и т.д.)

При входе в систему пользователь подключает к компьютеру этот носитель, и система считывает с него идентификационную и аутентификационную информацию пользователя. Далее аутентификация осуществляется, как было описано выше.

Аутентификационный ключ гораздо длиннее, чем пароль, поэтому подобрать такой ключ практически невозможно.

Однако угроза компрометации аутентификационных данных по-прежнему остается актуальной.

Описываемый механизм аутентификации, как правило, используется в совокупности с предыдущим. При этом пользователь, входя в систему, должен не только предъявить компьютеру носитель, но и ввести соответствующий этому носителю пароль (например, числовой пин-код).

Общие сведения

Аутентификация с использованием внешних носителей информации

Основной угрозой при использовании описываемого механизма аутентификации является угроза кражи носителя аутентификационных данных с последующим его копированием и подбором пароля на доступ к ключу.

Если аутентификационные данные выбираются случайно и формат их хранения на носителе не содержит проверочных полей (контрольных сумм и т.п.), оффлайн-подбор пароля на доступ к носителю аутентификационных данных невозможен – нарушитель просто не сможет сформулировать критерий, позволяющий отличить правильно расшифрованные аутентификационные данные от неправильно расшифрованных, и, следовательно, правильный пароль от неправильного.

Во многих реализациях такого механизма аутентификации применяются следующие дополнительные меры защиты:

- защита ключевого носителя от копирования;
- блокировка или уничтожение информации после определенного количества неудачных попыток ввода пин-кода (пароля) на доступ к ключу.

Однако эти меры защиты не всегда применимы (например, для touch memory или обычных Memory Card). Для Smart Card такой проблемы не существует.

Общие сведения

Аутентификация с использованием внешних носителей информации

В целом использование для аутентификации пользователей не только паролей, но и еще и внешних носителей информации позволяет заметно повысить защищенность ОС.

Но, с другой стороны, при этом у администраторов и пользователей возникает целый ряд проблем:

- проблема генерации ключей;
- проблема рассылки ключей;
- проблема смены ключей;
- проблема потерянных ключей.

Общие сведения

Биометрическая аутентификация

Каждый человек обладает своим неповторимым набором биометрических характеристик, к которым относятся:

- отпечатки пальцев;
- геометрическая форма руки;
- узор радужной оболочки глаза;
- рисунок сетчатки глаза;
- геометрическая форма и размеры лица;
- тембр голоса;
- геометрическая форма и размеры уха и др.

При такой схеме аутентификации угрозы компрометации и подбора аутентификационных данных перестают быть актуальными – подделать биометрические характеристики человека, как правило, настолько сложно и дорого, что затраты злоумышленника на проникновение в защищенную систему превысят выгоды от такого проникновения. Таким образом, данный механизм аутентификации создает практически непреодолимую защиту на этапе аутентификации.

Общие сведения

Биометрическая аутентификация

Основные достоинства аутентификации пользователей по их биометрическим характеристикам:

- трудность фальсификации этих признаков;
- высокая достоверность аутентификации из-за уникальности таких признаков;
- неотделимость биометрических признаков от личности пользователя.

Для сравнения аутентификации пользователей на основе тех или иных биометрических характеристик применяются оценки вероятностей ошибок первого и второго рода.

Вероятность ошибки первого рода (отказ доступа в систему легального пользователя) составляет 10^{-6} – 10^{-3} .

Вероятность ошибки второго рода (допуска к работе в системе незарегистрированного пользователя) в современных системах биометрической аутентификации составляет 10^{-5} – 10^{-2} .

Общий недостаток таких систем – более высокая стоимость по сравнению с другими механизмами аутентификации.

Общие сведения

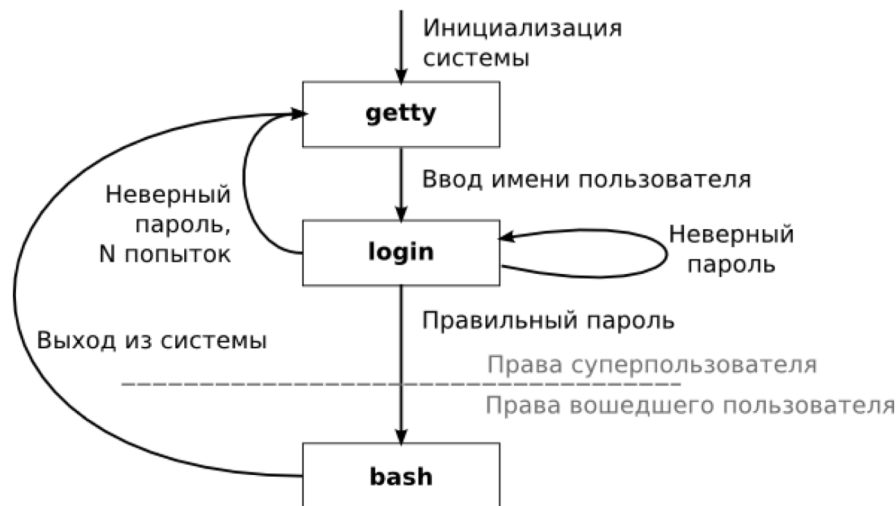
Биометрическая аутентификация

Наряду с этим, практическая реализация данного механизма аутентификации неизбежно создает ряд проблем, к основным из которых относятся следующие:

- для псевдопользователей должен поддерживаться альтернативный механизм;
- при двух последовательных входах в систему одного и того же человека его биометрические характеристики никогда в точности не совпадают (отсюда проистекают ошибки первого и второго рода);
- большинство биометрических характеристик человека постепенно меняются со временем;
- биометрические характеристики человека могут испытывать резкие кратковременные изменения.

Аутентификация в UNIX

Классический подход (до 1995 г.)



/etc/passwd

```
root:lZTB0KTrSKy8M:0:0:root:/root:/bin/sh
daemon:x:1:1:daemon:/usr/sbin:/bin/false
bin:x:2:2:bin:/bin:/bin/false
<...>
john:H5ned8EVlvank:101:101:./home/john:/usr/bin/csh
bill:7qeFjndagetZk:102:102:./home/bill:/bin/sh
```

```
useradd
userdel
usermod
groupadd
groupdel
groupmod
```

/etc/shadow

```
root:13$/Xhw3kaR$Vif2djTL4aQshu8aKfk10/:12545:0:99999:7:::
daemon*:12545:0:99999:7:::
bin*:12545:0:99999:7:::
<...>
john:13$8011AVB5$6AdyTstdHpsSTsewiac801:13283:0:31:3:14:13514:
bill:13$UdKpet5.$ssoFdtbe21qd.FL1gj19/0:13286:0:31:3:14:13514:
```

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

В 1995 году компанией SUN была предложена идея отделить программы от механизма аутентификации. Эта система получила название PAM (Pluggable Authentication Modules), что по-русски означает Подгружаемые Модули Аутентификации.

В качестве автономной инфраструктуры PAM впервые появился в Linux-PAM, разработанной в Red Hat Linux 3.0.4 в августе 1996 года. В настоящее время PAM поддерживается в следующих UNIX-like системах:

- AIX,
- FreeBSD,
- DragonBSD,
- NetBSD,
- HP-UX,
- Linux,
- MacOS X,
- Solaris.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)



Теперь если программа (в частности login) желает произвести аутентификацию пользователя она больше не занимается этим, а обращается к PAM'у с соответствующей просьбой. Последний выполняет все проверки и докладывает вызвавшему его о результатах - пускать или не пускать пользователя в систему. Все заботы о выборе алгоритма и особенностях аутентификации теперь лежат на PAM'е.

Программные модули, входящие в подсистему аутентификации UNIX-системы, делятся на три основные группы:

- клиенты (приложения и демоны), пользующиеся услугами PAM (login, sshd, passwd, rlogin, telnetd, ftpd и т. п.);
- программные модули (pam_*.so), предоставляющие услуги PAM. Обычно эти модули представляют собой библиотеки, размещаемые в директории /lib/security;
- файлы-сценарии (располагаются в /etc/pam.d) для использования PAM из конкретных программ. Имя файла-сценария совпадает с именем программы, его использующей.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Помимо файлов-сценариев для некоторых модулей могут использоваться дополнительные файлы конфигурации. Все они расположены в каталоге `/etc/security` и каждый файл предназначен для конкретной группы настроек:

- `time.conf`,
- `pam_env.conf`,
- `limits.conf`,
- `access.conf`,
- `group.conf`,
- `console.perms`.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

С точки зрения клиента аутентификации его взаимодействие с PAM реализуется посредством так называемых примитивов — PAM API вызовов, передающих управление соответствующим модулям PAM.

Поддерживаются шесть основных примитивов, сгруппированных в четыре подсистемы:

- `ram_authenticate` (подсистема `auth`) — аутентифицировать пользователя;
- `ram_setcred` (подсистема `auth`) — авторизовать пользователя (установить UID, идентификаторы групп, квоты ресурсов и т. д.);
- `ram_acct_mgmt` (подсистема `account`) — проверить, доступна ли учетная запись пользователя для авторизации (не устарел ли пароль, не заблокирована ли учетная запись и т. п.);
- `ram_open_session` (подсистема `session`) — начать сеанс работы пользователя с операционной системой;
- `ram_close_session` (подсистема `session`) — завершить сеанс работы пользователя с операционной системой;
- `ram_chauthtok` (подсистема `password`) — назначить пользователю новые аутентификационные данные.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Клиентские программы, обращающиеся к PAM, могут не знать о том, какой метод был использован при аутентификации некоторого конкретного пользователя. Вся техническая реализация процедуры аутентификации пользователя инкапсулирована внутри PAM модуля, клиенту выдается лишь самая общая информация о результатах выполнения PAM модуля того или иного примитива.

Каждый модуль PAM должен содержать функции-обработчики примитивов хотя бы одной подсистемы. Когда клиентская программа вызывает тот или иной примитив, PAM обращается к функциям-обработчикам данного примитива одного или нескольких своих модулей.

Важно отметить, что вся значимая функциональность PAM сосредоточена в модулях, сама система PAM не выполняет никаких действий, связанных с аутентификацией, а всего лишь вызывает в определенной последовательности заданные функции заданных модулей и принимает решение на основании возвращаемых ими результатов. Конкретный порядок того, какие функции каких модулей должны вызываться и как должны интерпретироваться результаты их вызова, определяется содержимым файлов-сценариев PAM.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Структура файла-сценария

В каждом файле-сценарии содержится последовательность строк, определяющих, какие PAM-модули должны использоваться клиентом и каким образом это должно осуществляться. Конфигурация PAM, используемая клиентскими программами по умолчанию, описывается в файле-сценарии `/etc/pam.d/other`.

Рассмотрим файл-сценарий для приложения `login`, находящийся по адресу `/etc/pam.d/login`. Его содержимое может выглядеть, например, так:

```
%PAM-1.0
auth requisite /lib/security/pam_unix.so nullok #set_secure
auth required /lib/security/pam_securetty.so
auth required /lib/security/pam_env.so
auth required /lib/security/pam_mail.so
account required /lib/security/pam_unix.so
password required /lib/security/pam_unix.so strict=false
session required /lib/security/pam_unix.so none # debug or trace
session required /lib/security/pam_limits.so
```

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Структура файла-сценария

Каждая строка файла соответствует одному модулю PAM и имеет вид:

type control path arguments.

■ Поле **type** описывает тип модуля. Каждой подсистеме примитивов PAM соответствует похожий по имени тип модуля PAM. При вызове клиентской программой примитива, принадлежащего определенной подсистеме, будут последовательно вызваны соответствующие функции-обработчики из всех модулей PAM соответствующего типа, при этом порядок вызова функций соответствует порядку перечисления модулей в файле-сценарии. Тип модуля должен быть одним из следующих:

- **auth**: Модуль проверяет наличие пользователя в системе, спрашивает его имя, разрешает или нет доступ в ту или иную группу (независимо от записей в файле `/etc/groups`) и вообще способен давать привилегии (конечно специально предназначенные для этого).
- **account**: Модуль не занимается аутентификацией, а позволяет контролировать распределение ресурсов системы для тех или иных пользовательских бюджетов.
- **session**: Модуль связан с событиями, которые могут происходить перед тем как пользователь получит доступ к той или иной службе. Например ведение записей в системных журналах.
- **password**: Модуль, как следует из названия, занимается непосредственно проверкой паролей на подлинность, на слабость и т.д.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Структура файла-сценария

■ Поле **control** описывает порядок интерпретации системой PAM результата обращения к данному модулю. Если функция-обработчик возвращает значение PAM.SUCCESS, считается, что модуль отработал успешно, в противном случае считается, что модуль сообщил об ошибке. Эти сведения обрабатываются в зависимости от значения поля control следующим образом:

- **requisite** (необходимый) — если модуль сообщает об ошибке, выполнение текущего примитива немедленно прерывается, клиентская программа получает сообщение об ошибке;
- **required** (требуемый): если модуль сообщает об ошибке, клиентская программа получает сообщение об ошибке, но выполнение примитива продолжается (возможно, обнаружатся и другие ошибки, клиенту будет полезно знать обо всех);
- **sufficient** (достаточный) — если модуль отработал успешно, выполнение примитива считается успешно завершенным (если не провалилась проверка на предшествующих модулях с флагом required), функции-обработчики последующих модулей не вызываются;
- **optional** (дополнительный) — модуль реализует второстепенные функции, не влияющие на общий статус выполнения запроса, статус обращения к модулю игнорируется. Например, модуль может ограничиться выводом на экран предупреждением о слабости вашего пароля.

Кроме перечисленных значений, поле **control** может также принимать два специальных значения **include** и **substack**, указывающих, что при интерпретации файла-сценария на место данной строки должны быть последовательно подставлены все строки типа **type** из файла-сценария по адресу, указанному в поле **path**. Поле **arguments** в этом случае игнорируется.



Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Структура файла-сценария

- Поле **path** содержит путь к соответствующему модулю PAM.
- Поле **arguments** содержит текстовую строку произвольного вида, которая будет передана данному модулю в качестве параметра.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Прежде чем рассмотреть содержательные примеры файлов-сценариев, рассмотрим функциональное назначение некоторых стандартных PAM модулей.

■ **pam_deny.so**: Тип любой. Всегда перекрывает доступ.

■ **pam_warn.so**: Тип auth и password. Ведет записи в системных журналах, например, при смене пароля.

■ **pam_unix.so**: Тип любой. Реализует UNIX-аутентификацию по умолчанию (вычисляет хеш-функцию пароля, заданную текущей конфигурацией операционной системы и сравнивает со значением, хранящимся в /etc/shadow). Практически всегда включается в скрипт аутентификации по умолчанию, включаемый в конфигурационные файлы всех клиентов через директиву include.

■ **pam_cracklib.so**: Тип password. Проверяет пароль на стойкость, не является ли он, например, палиндромом (это не обязательно при использовании модуля pam_unix.so). Полезен для программ, задающих пароли.

■ **pam_nologin.so**: Тип auth. Стандартная реакция на наличие файла /etc/nologin. Когда он присутствует, в систему может зайти только root, а остальным будет выдано на экран содержимое этого файла.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

■ **pam_access.so**: Тип любой. Модуль обеспечивает управление входом в систему. Этот модуль может использоваться для принятия решения о том, каким пользователям разрешен вход в систему. Так как PAM имеет средства аутентификации по сети, то контролируется не только кто может или не может зайти, но и откуда. По умолчанию правила управления доступом берутся из файла конфигурации `/etc/security/access.conf`.

■ **pam_limits.so**: Тип session. Позволяет индивидуально или для группы пользователей ограничить: размер core-файла, максимальный допустимый размер файла, максимальное количество открытых файлов, запущенных процессов, сколько раз можно одновременно зайти в систему и т.д. Использует дополнительный файл конфигурации `/etc/security/limits.conf`.

■ **pam_umask.so**: Тип session. Модуль служит для установки маски создания файла для текущего окружения. Маска влияет на разрешения по умолчанию, назначаемые вновь создаваемым файлам.

■ **pam_pwcheck.so**: Тип любой. Модуль служит для проверки стойкости пароля.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Примеры

- Если для клиентской программы не указан файл-сценарий в `/etc/pam.d`, то в этом случае используется файл-сценарий по умолчанию `/etc/pam.d/other`.

#

по умолчанию; запрещает любой доступ

#

auth	required	pam_deny.so
auth	required	pam_warn.so
account	required	pam_deny.so
password	required	pam_deny.so
password	required	pam_warn.so
session	required	pam_deny.so

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Примеры

- Правила PAM, эквивалентные стандартным правилам безопасности UNIX.

#

стандартные минималистичные правила UNIX

#

auth	required	pam_unix.so
account	required	pam_unix.so
password	required	pam_unix.so
session	required	pam_unix.so

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Примеры

■ Секция password в файле `/etc/pam.d/passwd` определяет хорошие требования, предъявляемые к новым паролям.

```
#  
# retry=3      три подхода для установки нового пароля  
# minlen=10    требуется как минимум десять символов  
# ucredit=-1   как минимум один символ в верхнем регистре  
# lcredit=0    любое количество символов в нижнем регистре  
# dcredit=-2   как минимум две цифры  
# ocredit=-1   как минимум один неалфавитный символ  
# password required pam_cracklib.so retry=3 minlen=10 \  
#             ucredit=-1 lcredit=0 dcredit=-2 ocredit=-1  
  
#  
# Модуль pam_cracklib лишь проверяет пароли, но не сохраняет их.  
# Для этого нам потребуется еще стандартный модуль pam_unix.  
# Параметр use_authok означает, что модуль pam_unix не будет  
# запрашивать пароль, а будет как раз использовать пароль,  
# предоставленный модулем pam_cracklib.  
# nullok — можно использовать пустые пароли  
#  
password required pam_unix.so use_authok nullok
```

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Примеры

- Файл `/etc/pam.d/sshd` содержит правила безопасности для SSH-соединений.

```
auth required pam_unix.so
auth required pam_nologin.so
account required pam_unix.so
account required pam_access.so
session required pam_limits.so
session required pam_unix.so
session optional pam_umask.so
password requisite pam_pwcheck.so cracklib
password required pam_unix.so use_authtok
```

Модуль `pam_access.so` использует дополнительные проверки, указанные в дополнительном файле конфигурации `/etc/security/access.conf`:

```
+ : ALL : 192.168.
+ : remoteOleg : ALL
- : ALL : ALL
```

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Таким образом, архитектура PAM позволяет администратору UNIX-системы гибко настраивать подсистему аутентификации в соответствии с реальными потребностями конкретной вычислительной сети и конкретного экземпляра операционной системы.

Все, что нужно сделать администратору, — установить в системе необходимые модули PAM и обеспечить корректное взаимодействие между ними. В результате подсистема аутентификации может быть легко адаптирована к самым разным технологиям аутентификации, принятым в конкретной организации.

В частности, на базе PAM может быть реализована аутентификация с использованием смарт-карт или биометрическая аутентификация. Для этого достаточно, чтобы разработчик аппаратного устройства, используемого для аутентификации, поставлял вместе со своим устройством соответствующий модуль PAM.

Аутентификация в UNIX

Pluggable Authentication Modules (PAM)

Источники информации

- Официальная документация PAM: www.kernel.org/pub/linux/libs/pam
- Описание конфигурационных файлов: www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/sag-configuration-file.html
- Описание повсеместных модулей PAM: www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/sag-module-reference.html

Аутентификация в Windows

В Windows задачи идентификации, аутентификации и авторизации пользователей решаются специальной *подсистемой аутентификации*. Подсистема аутентификации Windows делится на три уровня — верхний, средний и нижний. Средний уровень подсистемы аутентификации пользуется услугами нижнего уровня и предоставляет услуги верхнему.

Верхний уровень подсистемы аутентификации Windows включает в себя процесс аутентификации winlogon.exe и так называемые *провайдеры аутентификации (поставщики учетных записей)* — заменяемые библиотеки, реализующие большую часть высокоуровневых функций процесса аутентификации.

Процесс Winlogon представляет собой обычный процесс, выполняющийся от имени псевдопользователя SYSTEM. Данный процесс автоматически запускается при старте операционной системы и остается активным до выключения питания или перезагрузки. При аварийном завершении Winlogon происходит аварийное завершение работы всей операционной системы («синий экран»). Таким образом, подменить Winlogon в процессе функционирования операционной системы практически невозможно.

Аутентификация в Windows

Winlogon является доверенным процессом, отвечающим за управление взаимодействий с пользователем, связанных с безопасностью. Им координируются вход в систему, запуск первого процесса пользователя при входе в систему, обработка выхода из системы и управление рядом других операций, относящихся к безопасности, включая запуск LogonUI для ввода паролей при входе в систему, изменении паролей и блокировке и разблокировке рабочей станции.

Процесс Winlogon должен гарантировать, что операции, связанные с безопасностью, невидимы любым другим активным процессам. Например, Winlogon гарантирует, что не пользующийся доверием процесс не может получить управление рабочим столом в ходе одной из таких операций, получив тем самым доступ к паролю.

Winlogon при получении имени и пароля учетной записи пользователя зависит от установленных в системе поставщиков учетных записей. Эти поставщики являются COM-объектами, которые находятся внутри DLL-библиотек. Исходными поставщиками являются authui.dll, SmartcardCredentialProvider.dll и FaceCredentialProvider.dll, и они поддерживают пароль, PIN аутентификации смарткарты и аутентификацию с распознаванием лица. Разрешение установки других поставщиков учетных данных позволяет Windows использовать различные механизмы идентификации пользователей.

Аутентификация в Windows

Чтобы защитить адресное пространство Winlogon от ошибок поставщиков учетных данных, которые могут привести к сбою процесса Winlogon (что, в свою очередь, приведет к системному сбою, поскольку Winlogon считается критическим системным процессом), для фактической загрузки поставщиков учетных данных и демонстрации пользователю Windows-интерфейса входа в систему используется отдельный процесс LogonUI.exe. Этот процесс запускается по запросу, как только Winlogon требуется присутствие пользовательского интерфейса, а выход из него осуществляется после завершения нужного действия. Это также позволяет Winlogon просто перезапустить новый процесс LogonUI в случае сбоя, возникшего по какой-либо причине.

Аутентификация в Windows

Верхний уровень аутентификации

Вход локального пользователя в систему обычно выполняется в Windows следующим образом.

- Провайдер аутентификации получает от пользователя идентификационную и аутентификационную информацию. В стандартной конфигурации операционной системы в качестве идентификационной информации используется текстовое имя, а в качестве аутентификационной информации — текстовый пароль. Также возможно применение для аутентификации внешних носителей ключевой информации или биометрических характеристик пользователя.
- Провайдер аутентификации генерирует запрос на аутентификацию, передавая необходимые данные на средний уровень подсистемы аутентификации с помощью системного вызова `LsaLogonUser` или одной из более высокоуровневых программных оберток этого системного вызова. Если аутентификация прошла успешно, создается маркер доступа пользователя.
- Если маркер доступа пользователя создан успешно, провайдер аутентификации осуществляет авторизацию пользователя, запуская процесс `userinit.exe` от имени аутентифицированного пользователя. Для этого используется системный вызов `CreateProcessAsUser`, который отличается от вызова `CreateProcess` только тем, что запускаемому процессу назначается маркер доступа, отличный от маркера доступа процесса-родителя. В данном случае процессу `userinit` назначается только что созданный маркер доступа авторизуемого пользователя.
- Процесс `userinit` загружает индивидуальные настройки пользователя из его профиля, запускает программу-оболочку пользователя (чаще всего это Проводник Windows) и после этого завершает работу.

Аутентификация в Windows

Средний уровень аутентификации

В средний уровень подсистемы аутентификации Windows входит *локальный распорядитель безопасности (local security authority, LSA)* и так называемые *пакеты, аутентификации* — заменяемые библиотеки, реализующие большую часть низкоуровневых функций аутентификации.

Локальный распорядитель безопасности представляет собой сервисный процесс lsass.exe, выполняющийся от имени псевдопользователя SYSTEM. Аварийное завершение LSA приводит к аварийному завершению работы всей операционной системы.

Так же, как и Winlogon, LSA передоверяет большинство своих функций заменяемым библиотекам. Стандартная схема аутентификации Windows NT обслуживалась пакетом аутентификации MSV 1.0 (msvl_0.dll), а начиная с Windows 2000, стандартным является пакет аутентификации Kerberos.

Аутентификация в Windows

Средний уровень аутентификации

Пакет аутентификации осуществляет аутентификацию пользователя в процессе обработки системного вызова LsaLogonUser. Аутентификация производится следующим образом.

- Пакет аутентификации получает от верхнего уровня подсистемы аутентификации имя и пароль пользователя и генерирует образ пароля.
- Используя услуги нижнего уровня подсистемы аутентификации, пакет аутентификации получает информацию, необходимую для проверки пароля, и проверяет пароль. Проверка пароля может вестись как путем простого сравнения хеш-образа введенного пароля с эталонным хеш-образом (протоколы LanManager, NTLM), так и путем более сложных криптографических процедур (Kerberos).
- Если введенный пароль признан корректным, LSA получает от нижнего уровня подсистемы аутентификации информацию о том, может ли данный пользователь начинать в данный момент работу с данной рабочей станцией (не устарел ли пароль, не заблокирован ли бюджет пользователя и т. д.).
- В случае положительного результата проверки LSA формирует маркер доступа пользователя, получая необходимую информацию от нижнего уровня подсистемы аутентификации.
- LSA передает сформированный маркер доступа верхнему уровню подсистемы аутентификации.



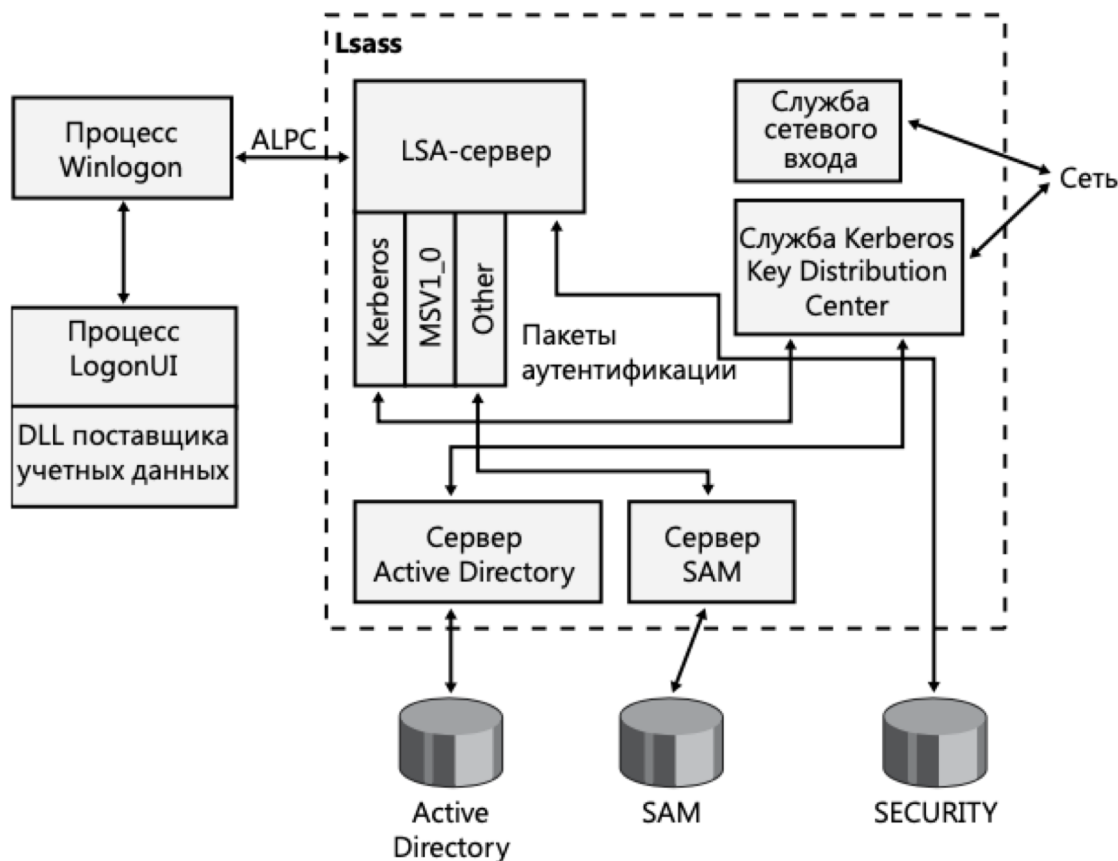
Аутентификация в Windows

Нижний уровень аутентификации

Нижний уровень подсистемы аутентификации Windows отвечает за хранение в системе учетной информации о пользователях, в том числе и эталонных образов паролей. При аутентификации пользователя нижний уровень подсистемы аутентификации передает среднему уровню эталонный образ пароля пользователя, а при авторизации — список групп и привилегий пользователя.

Аутентификация в Windows

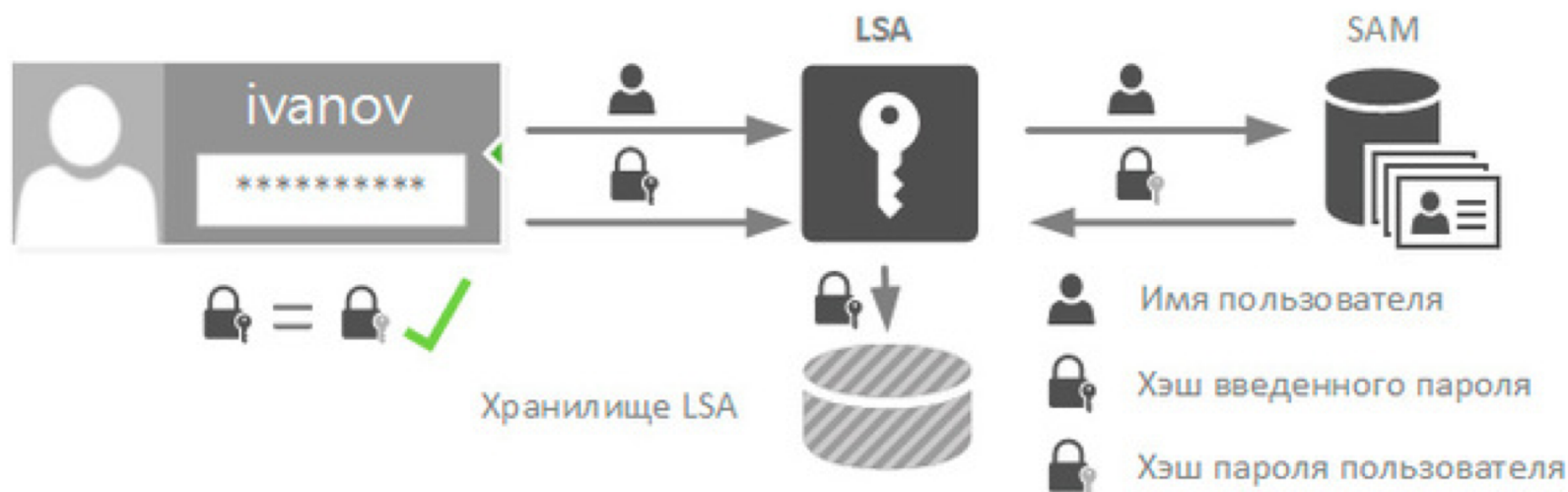
Взаимодействие между уровнями аутентификации, участвующими во входе в систему, показано на рисунке.



LSASS - Local Security Authority Subsystem Service
LSA - Local Security Authority
ALPC - Advanced Local Procedure Call
SAM - Security Account Manager

Аутентификация в Windows

Более упрощенно схема локальной аутентификации выглядит следующим образом.





Аутентификация в Windows

Аутентификация при удаленном входе

Аутентификация при удаленном входе в систему осуществляется в целом по той же схеме за исключением того, что на верхнем уровне вместо процесса Winlogon может выступать произвольная пара клиент + сервер. Существует специальный интерфейс SSPI (Security Support Provider Interface), обеспечивающий взаимодействие приложений Windows с LSA в ходе аутентификации. В Windows поддерживается несколько стандартных провайдеров сетевой аутентификации.

Аутентификация в Windows

Аутентификация при удаленном входе

LAN Manager (LM)

Протокол LAN Manager возник на заре зарождения локальных сетей под управлением Windows и впервые был представлен в Windows 3.11 для рабочих групп, откуда переключался в семейство Windows 9.x. Мы не будем рассматривать этот протокол, так как в естественной среде он уже давно не встречается, однако его поддержка, в целях совместимости, присутствует до сих пор. И если современной системе поступит запрос на аутентификацию по протоколу LM, то, при наличии соответствующих разрешений, он будет обработан.

Почему этот протокол в настоящее время не используется? Попробуем разобраться. Прежде всего разберемся, каким образом создается хэш пароля для работы с протоколом LM. Не вдаваясь в подробности обратим внимание на основные ограничения:

- пароль регистронезависимый и приводится к верхнему регистру;
- максимальная длина пароля - 14 символов, более короткие пароли дополняются при создании хэша нулями;
- пароль делится пополам и для каждой части создается свой хэш по алгоритму DES.



Аутентификация в Windows

Аутентификация при удаленном входе

LAN Manager (LM)

Исходя из современных требований к безопасности, можно сказать, что LM-хэш практически не защищен и будучи перехвачен очень быстро расшифровывается. Хотя прямое восстановление хэша невозможно, однако в силу простоты алгоритма шифрования возможен подбор соответствующего паролю образа за предельно короткое время.

А теперь самое интересное. LM-хэш, в целях совместимости, создается при вводе пароля и хранится в системах по Windows XP включительно. Это делает возможной атаку, когда системе целенаправленно присылают LM-запрос и она его обрабатывает. Избежать создания LM-хэша можно, изменив политику безопасности, или, используя пароли длиннее 14 символов. В системах, начиная с Windows Vista и Server 2008, LM-хэш по умолчанию не создается.

Аутентификация в Windows

Аутентификация при удаленном входе

NT LAN Manager (NTLM)

Новый протокол аутентификации появился в Windows NT 4.0 и благополучно, с некоторыми изменениями, дожил до наших дней. А до появления Kerberos в Windows 2000 был единственным протоколом аутентификации в домене NT.

Сегодня протокол NTLM, а точнее, его более современная версия NTLMv2, применяется для аутентификации компьютеров рабочих групп. В доменных сетях Active Directory по умолчанию применяется Kerberos, однако если одна из сторон не может применить этот протокол, то по согласованию могут быть использованы NTLMv2, NTLM и даже LM.

Принцип работы NTLM имеет много общего с LM и эти протоколы обратно совместимы, но есть и существенные отличия. NTLM-хэш формируется на основе пароля длиной до 128 символов по алгоритму MD4, пароль регистрозависимый и может содержать не только ACSII символы, но и Unicode, что существенно повышает его стойкость по сравнению с LM.

Аутентификация в Windows

Аутентификация при удаленном входе NT LAN Manager (NTLM)

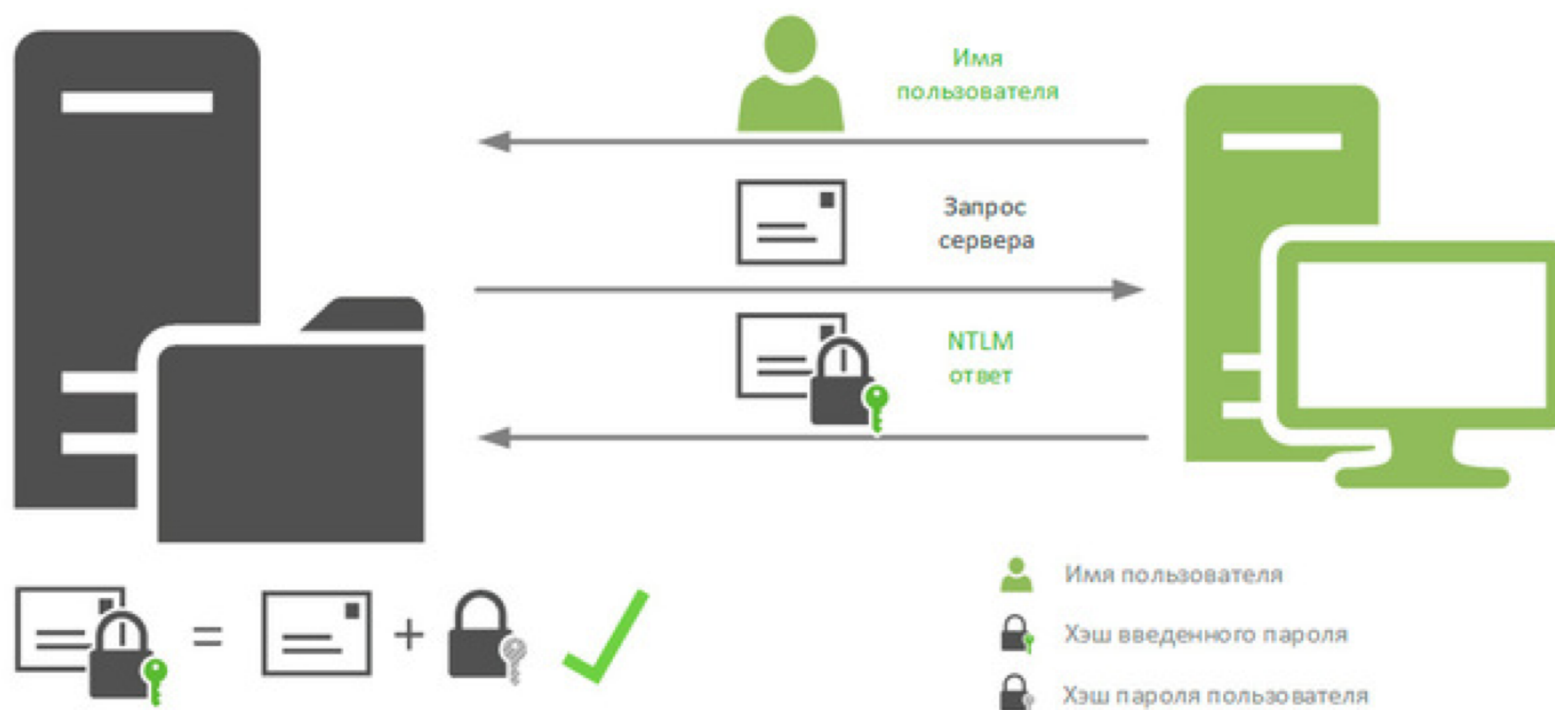
Алгоритм аутентификации выглядит в общих чертах следующим образом.

- Клиент направляет серверу имя пользователя в открытом виде (в NTLM идентификационная информация пользователя не считается секретом).
- В ответ сервер генерирует случайное число от 0 до 65535, называемое *запросом сервера*, и высылает его клиенту.
- Клиент в свою очередь зашифровывает данный запрос по алгоритму DES, используя в качестве ключа NTLM-хеш пароля пользователя. Однако, несмотря на то, что NTLM-хэш 128-битный, в силу технических ограничений используется 40 или 56 битный ключ (хэш делится на три части и каждая часть шифрует запрос сервера отдельно). Зашифрованный хэшем пароля запрос сервера называется *ответом NTLM*. Ответ NTLM передается обратно серверу.
- Сервер извлекает из хранилища SAM хэш пароля того пользователя, чье имя было ему передано и выполняет аналогичные действия с запросом сервера, после чего сравнивает полученный результат с ответом NTLM. Если результаты совпадают, значит пользователь клиента действительно тот, за кого себя выдает, и аутентификация считается успешной.

Аутентификация в Windows

Аутентификация при удаленном входе

NT LAN Manager (NTLM)

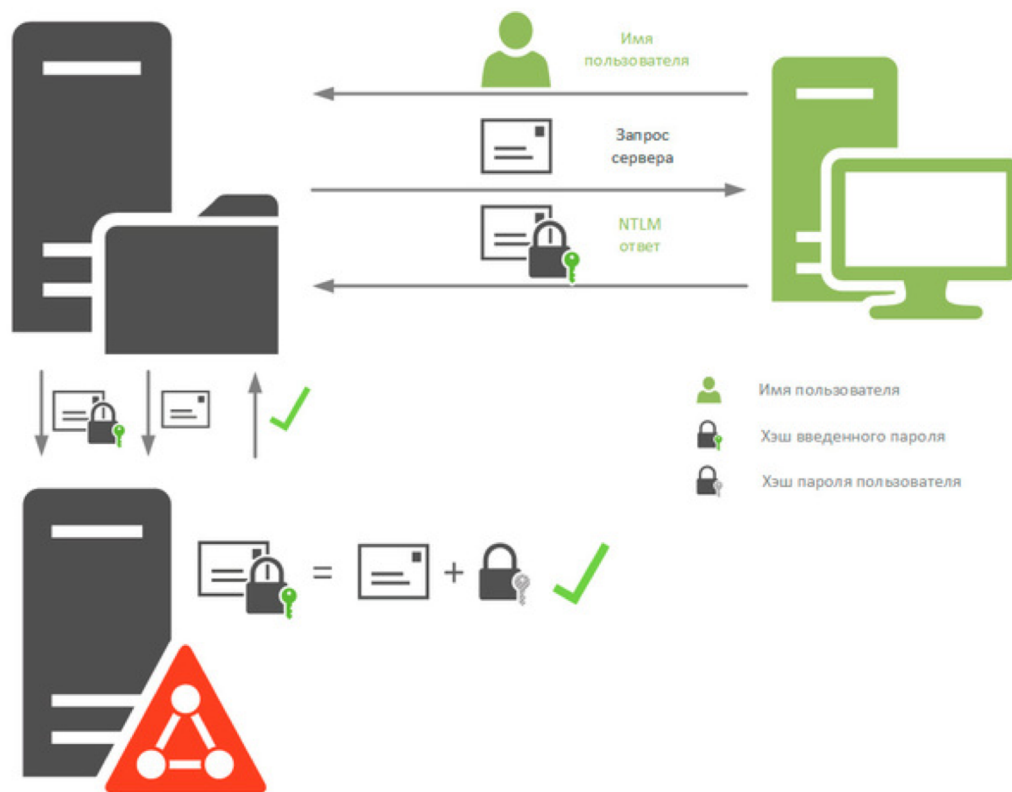


Аутентификация в Windows

Аутентификация при удаленном входе

NT LAN Manager (NTLM)

В случае доменной аутентификации процесс протекает несколько иначе. В отличие от локальных пользователей, хэши паролей которых хранятся в локальных базах SAM, хэши паролей пользователей хранятся на контроллерах доменов. При входе в систему LSA отправляет доступному контроллеру домена запрос с указанием имени пользователя и имени домена и дальнейший процесс происходит как описано выше.



Аутентификация в Windows

Аутентификация при удаленном входе

NT LAN Manager v2 (NTLMv2)

Осознавая, что протокол NTLM не соответствует современным требованиям безопасности, с выходом Windows 2000 Microsoft представила вторую версию протокола NTLMv2, который был серьезно доработан в плане улучшений криптографической стойкости и противодействия распространенным типам атак. Начиная с Windows 7 / Server 2008 R2 использование протоколов NTLM и LM по умолчанию выключено.

Криптостойкость данного алгоритма является актуальной и на сегодняшний день, известно только два случая взлома данного хэша. Один из них произведен компанией Symantec в исследовательских целях. Можно с уверенностью сказать, что в настоящий момент нет массовых инструментов для атак на NTLMv2, в отличие от NTLM, взломать который может любой вдумчиво прочитавший инструкцию студент.

Сразу рассмотрим схему с контроллером домена, в случае его отсутствия схема взаимодействия не меняется, только вычисления, производимые контроллером домена, выполняются непосредственно на сервере.

Аутентификация в Windows

Аутентификация при удаленном входе

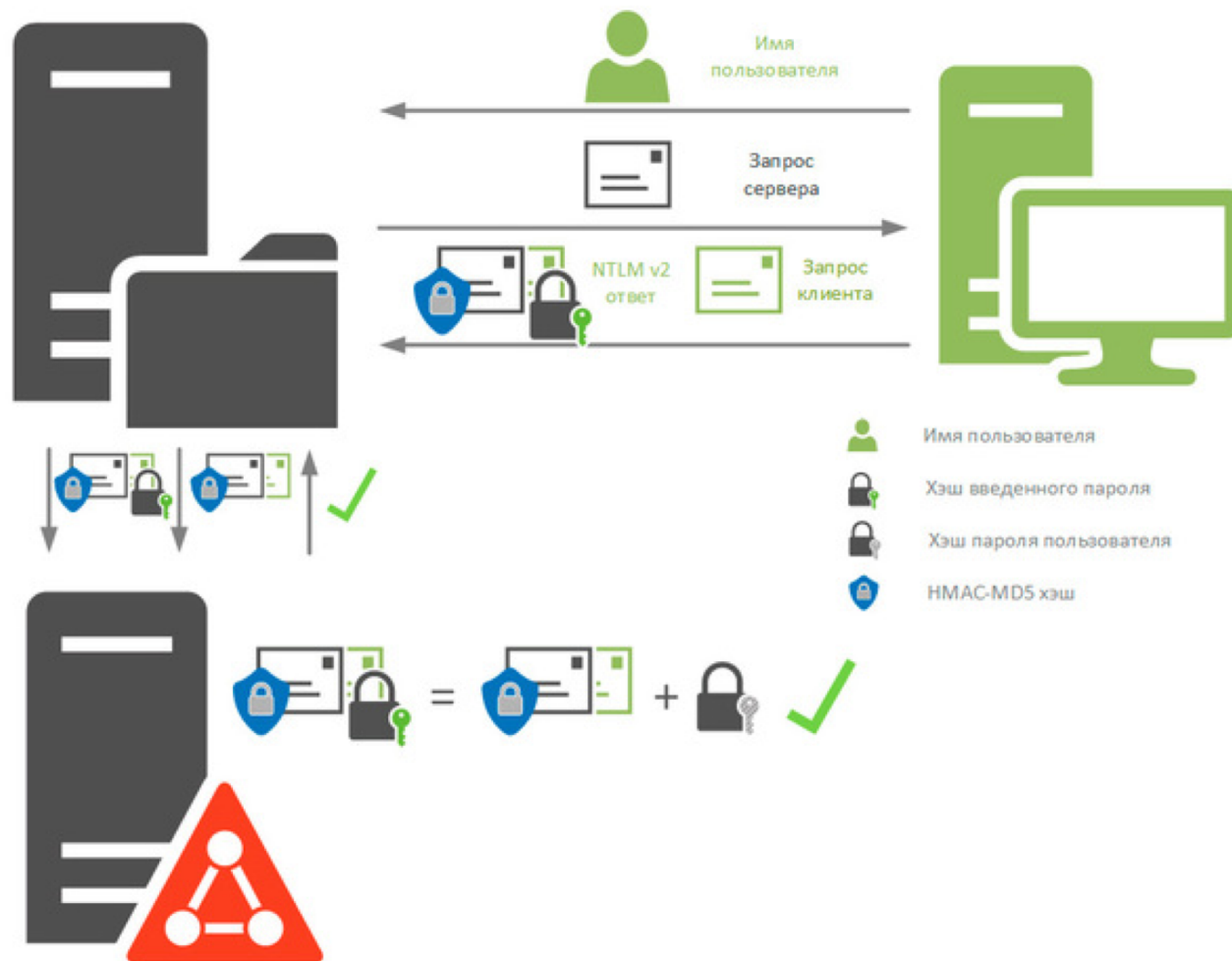
NT LAN Manager v2 (NTLMv2)

- Как и в NTLM, клиент при обращении к серверу сообщает ему имя пользователя и имя домена.
- В ответ сервер передает ему случайное число - запрос сервера.
- В свою очередь клиент генерирует также случайное число (куда, кроме прочего, добавляется метка времени) которое называется *запрос клиента*. Наличие метки времени позволяет избежать ситуации, когда атакующий первоначально накапливает перехваченные данные, а потом с их помощью осуществляет атаку.
- Запрос сервера объединяется с запросом клиента и от этой последовательности вычисляется HMAC-MD5 хэш. После чего от данного хэша берется еще один HMAC-MD5 хэш, ключом в котором выступает NTLM-хэш пароля пользователя. Получившийся результат называется NTLMv2-ответом и вместе с запросом клиента пересылается серверу.
- Сервер, получив NTLMv2-ответ и запрос клиента, объединяет последний с запросом сервера и также вычисляет HMAC-MD5 хэш, затем передает его вместе с ответом контроллеру домена.
- Контроллер домена извлекает из хранилища сохраненный хэш пароля пользователя и производит вычисления над HMAC-MD5 хешем запросов сервера и клиента, сравнивая получившийся результат с переданным ему NTLMv2-ответом. В случае совпадения серверу возвращается ответ об успешной аутентификации.

Аутентификация в Windows

Аутентификация при удаленном входе

NT LAN Manager v2 (NTLMv2)



Аутентификация в Windows

Аутентификация при удаленном входе

Kerberos

Протокол аутентификации Kerberos появился в Windows 2000. Он весьма сложен и детальное его рассмотрение выходит за рамки настоящего курса. Отметим лишь основные его достоинства и недостатки.

■ Основным достоинством протокола Kerberos является его чрезвычайно высокая стойкость. Даже перехватив весь трафик информационного взаимодействия всех участников процесса аутентификации, получить несанкционированный доступ к ресурсам любого из участников информационного обмена практически невозможно.

Особенно повышают защищенность Kerberos жесткие ограничения, которые данный протокол устанавливает на время аутентификации. Большинство данных, которые могут быть перехвачены нарушителем, устаревают спустя считанные минуты, некоторые данные могут сохранять актуальность несколько часов. В любом случае, современная вычислительная техника, включая суперкомпьютеры, не позволяет осуществлять взлом используемых криптографических алгоритмов за приемлемое время.

Аутентификация в Windows

Аутентификация при удаленном входе

Kerberos

■ Основным недостатком Kerberos является то, что аутентификация по этому протоколу требует некоторой подготовительной работы и не может быть выполнена произвольной парой клиент + сервер.

Как минимум, клиент и сервер должны выбрать сервера-посредника, которому они оба доверяют и который заранее осведомлен о некоторых характеристиках клиента и сервера.

Поэтому протокол Kerberos может эффективно применяться только в централизованно управляемых локальных сетях с априорно известными топологией и структурой.

Существуют модификации Kerberos для работы в Internet и даже для локальной аутентификации, но это, фактически, профанация — в этих режимах Kerberos не имеет никаких преимуществ по сравнению с более примитивными протоколами типа NTLM, но вычислительная сложность криптографических преобразований Kerberos существенно выше.

Аутентификация в Windows

Аутентификация при удаленном входе

Существуют также следующие стандартные провайдеры сетевой аутентификации:

■ **Negotiate** (поддерживается начиная с Windows 2000). Этот провайдер обеспечивает автоматический выбор провайдера между NTLM и Kerberos. В современных версиях Windows Negotiate выбирает NTLM лишь в тех случаях, когда использование Kerberos невозможно по техническим причинам. Как правило, приложения обращаются не к NTLM и не к Kerberos, а именно к Negotiate.

■ **Digest** (поддерживается начиная с Windows XP). Данный протокол аутентификации специально предназначен для веб-приложений. Подробно спецификации протокола изложены в RFC 2617. Функционально Digest похож на NTLM, для криптографических преобразований в Digest может использоваться поточный шифр RC4 с длиной ключа 40, 56 или 128 бит, а также DES либо Triple DES.

■ **Schannel** (поддерживается начиная с Windows NT 4.0 SP4). Этот провайдер поддерживает протоколы сетевой аутентификации TLS 1.0 и SSL 3.0, а также устаревший протокол PCT 1.0. К криптографическим преобразованиям, используемым Schannel, относятся RC2, RC4, DES, Triple DES, RSA, DHE, MD5, SHA.

Помимо перечисленных стандартных провайдеров, Windows может работать и с нестандартными провайдерами, созданными вне Microsoft. Интерфейсы, используемые провайдерами аутентификации, практически полностью документированы.



Аутентификация в Windows

Выше была изложена стандартная схема идентификации и аутентификации пользователя в Windows, которая применяется при использовании стандартных провайдеров и пакетов аутентификации.

Однако, поскольку и провайдеры, и пакеты аутентификации являются заменяемыми компонентами подсистемы аутентификации, администратор операционной системы может, установив нестандартный провайдер или пакет аутентификации, реализовать в Windows любую другую схему аутентификации. Для этого необходимо всего лишь разместить в системной директории Windows необходимые библиотеки и внести изменения в соответствующие ключи реестра.